Best Practices for Unstructured Grid Shock-Fitting

Peter L. McCloud*

ERC-Inc, Houston, TX 77058

Unstructured grid solvers have well-known issues predicting surface heat fluxes when strong shocks are present. Various efforts have been made to address the underlying numerical issues that cause the erroneous predictions. The present work addresses some of the shortcomings of unstructured grid solvers, not by addressing the numerics, but by applying structured grid best practices to unstructured grids. A methodology for robust shock detection and shock-fitting is outlined and applied to production-relevant cases. Results achieved by using the Loci-CHEM Computational Fluid Dynamics solver are provided.

I. Introduction

Traditionally, Computational Fluid Dynamics (CFD) solvers have used point-matched, structured grids for analyzing hypersonic cases. Two of these solvers, LAURA[1, 2] and DPLR[3–5] have long been considered the standard for hypersonic applications. The requirement to have point-matched structured grids can make grid generation tedious or nearly impossible when the geometry becomes complex. To address grid generation complexity, efforts are being made to create hypersonic CFD codes that utilize either structured overset grids[6, 7] or unstructured grids. Unstructured grids offer the promise of easier grid generation processes, but have inherent numerical complications when strong shocks are present. Satisfactory inviscid flux formulations to handle strong shocks are currently an ongoing area of research.

Two unstructured grid solvers currently being developed as the next generation of hypersonic solvers are US3D[8] and FUN3D.[9] While US3D has shown great promise for unstructured grids, it should be noted that the preferred grid topology for US3D is to use purely hexahedrals to take advantage of the higher order methods. US3D also requires that a cell connectivity map be created.

The present work sidesteps the underlying numerical issues and addresses the issues of strong shocks on unstructured grids by adopting structured grid solver best practices. This requires creating a more complex grid topology, but the flexibility of unstructured grid generation can easily incorporate the increased complexity. A methodology of performing unstructured shock-fitting on production-relevant cases was developed, tested and verified against test data.

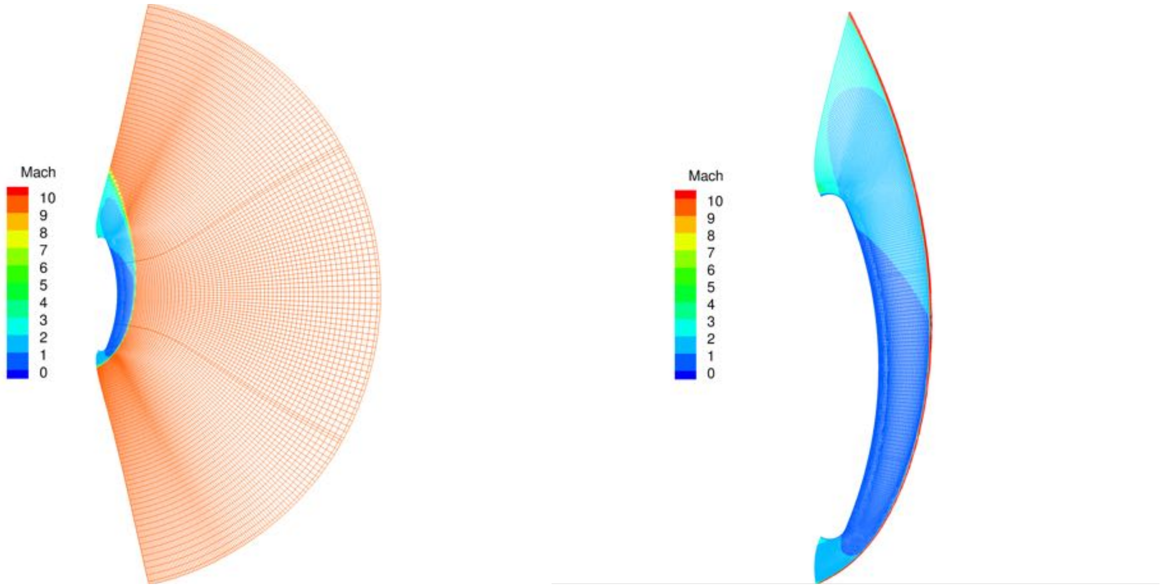
While the framework for performing the unstructured shock adaption was built to be solver agnostic, and could even work with structured grid solvers, the results shown were computed using the Loci-CHEM[10, 11] CFD solver. Loci-CHEM is an unstructured, finite-rate chemistry solver developed at Mississippi State University (MSU) by Dr. Ed Luke.

II. Structured Shock-Fitting Methodology

Structured grids for hypersonic problems are typically point-matched and the coordinates are labeled *i, j,* and *k*, with the *k* direction being the off-body direction. Once a solution is obtained on the initial grid, the shock-fitting process is started. The user specifies a percentage of the free stream Mach number. For DPLR, this is typically 95%. The adaption algorithm analyzes the *k*-lines and determines where the Mach number reaches the specified value, which represents the shock location. It should be noted that the identified points on all of the *k*-lines essentially represent a Mach iso-surface, which is utilized in the unstructured process. Once the shock surface is identified, the locations are typically smoothed relative to each other. Finally, the *k*-lines are then modified such that the k_{max} point lies just outside of the shock, with a minimal margin for maximum computational efficiency.

[*] Aerothermodynamics Engineer, JETS-EG3 NASA JSC, Lifetime Member

An example of the structured shock-fitting methodology on an axisymmetric heatshield forebody at a specified angle of attack is provided in figure 1. Figure 1a is the center-line Mach iso-contour for the initial unadapted grid, while the center-line Mach iso-contour for the final shock-adapted grid can be seen in figure 1b. Together, these two figures demonstrate how the *k*-lines are modified such that the endpoints lie just beyond the bow shock.



(a) Center-line Mach iso-contour for the initial unadapted grid.

(b) Center-line Mach iso-contour for the final shock-adapted grid.

Figure 1. Before and after shock-fitting results on an axisymmetric heatshield forebody at Mach 10 using structured grids.

The drawback to this methodology is that it does not allow for the existence of internal shocks or the presence of plumes in the flow field. As the algorithm follows the *k*-lines and encounters a location where the Mach number meets the criteria, even if not the bow shock, it will treat that feature as the bow shock and the process will fail.

III. Unstructured Shock-Fitting Methodology

The overall goal of the shock-fitting process is to create a grid with element faces that are aligned to the shock, decreasing the numerical noise. The present work concentrated on developing a shock-fitting methodology that leveraged structured grid best practices and made the process solver agnostic wherever possible in order for it to be applicable to a wide range of tools. The shock-fitting process developed for the present work is outlined in figure 2. The process is composed of an iterative loop composed of several sub-steps with the first loop starting from the initial solution.

The first step of the loop applies a shock detection algorithm to build a surface that represents the shock. Next, the surface is smoothed to better align the cell faces to the shock. To maximize the cell quality at the shock, the shock surface is then re-meshed. With the shock surface finalized, a prismatic layer is then extruded on both sides of the surface. The volume grid is then re-meshed using the prismatic layer as a new outer boundary. The final step of the loop is to solve for the updated solution. The loop is repeated until the solution achieves sufficient accuracy.

Implementation of the unstructured shock-fitting methodology is part of the *mesh_tools* tool suite developed at the NASA Johnson Space Center (JSC) Aerosciences Branch. The tool suite is a collection of Python scripts and modules that utilize the Visualization Tool Kit (VTK).[12] The VTK libraries provide efficient data structures and algorithms that maximize the performance of the Python scripts and utilize a pipeline process where filters are applied to data sets to obtain the desired results.

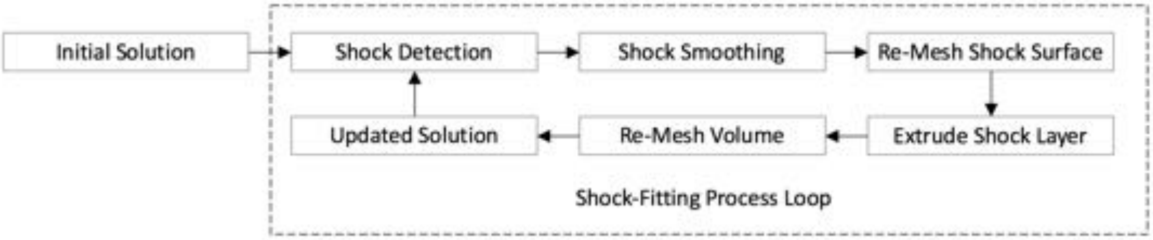


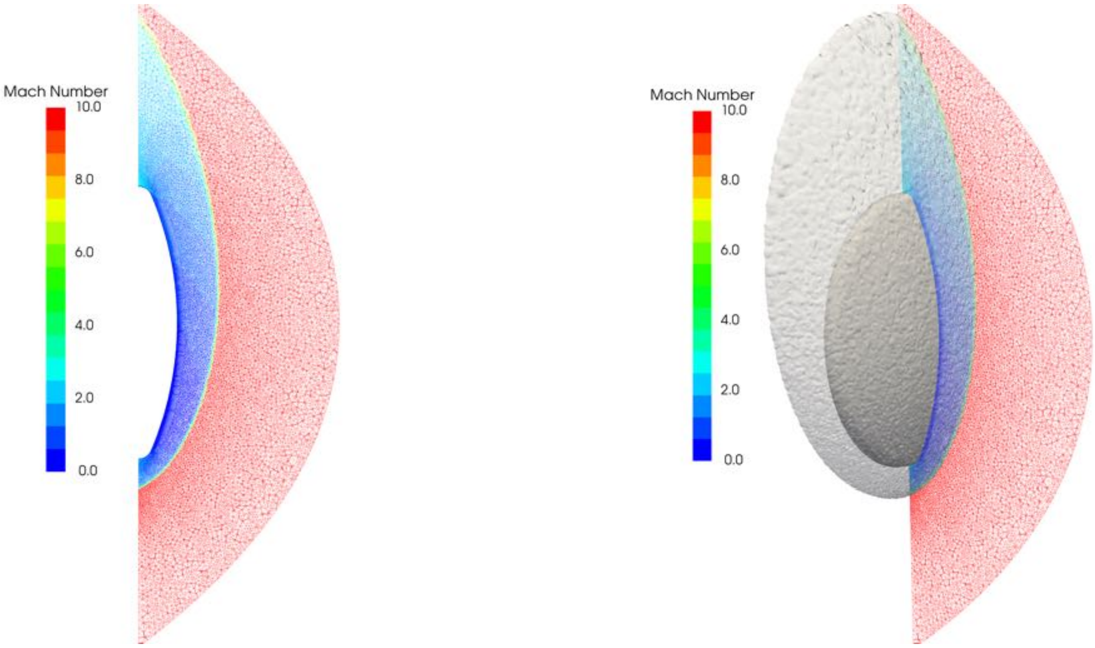
Figure 2. Shock-fitting process.

III.A. Initial Solution

As with structured grids, the first step in the process is to obtain an initial grid and solution. The present work utilized Altair's Hypermesh[13] to build a quad-dominant surface mesh and AFLR3[14, 15] to generate the volume mesh. The initial volume mesh consists of a prismatic layer grown from the viscous walls and the remaining volume is filled with tetrahedra. Obtaining the initial solution can sometimes be problematic due to carbuncles forming at the bow shock location. Typically, the best results are obtained by averaging the solution over of a number of iterations to account for the shock movement due to carbuncles.

III.B. Shock Detection

One of the main challenges with the unstructured shock-fitting is identifying the shock location efficiently without the connectivity information inherent in structured grids. As mentioned previously, the structured shock-fitting process is essentially identifying a Mach iso-contour. To start the unstructured shock-fitting process, the solution is read in and the Mach iso-contour is then found by simply applying the *vtkContourFilter*. For a typical smooth body hypersonic case, the resulting Mach iso-contour represents the bow shock. An example of the unstructured shock detection process performed on the axisymmetric heatshield forebody is found in figure 3. The initial center-line Mach iso-contour is seen in figure 3a. The resulting shock surface using the 95% freestream Mach number criteria is shown in figure 3b.



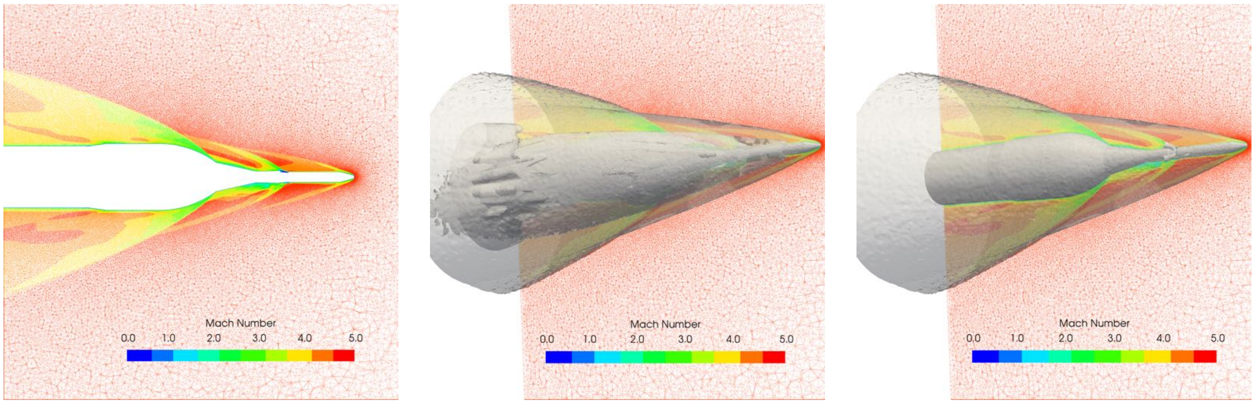
(a) Center-line Mach iso-contour for the initial unadapted, unstructured grid

(b) Center-line Mach iso-contour with a Mach iso-surface set to 95% of the freestream Mach number

Figure 3. Shock detection results for an axisymmetric heatshield forebody at Mach 10 using unstructured grids.

This simplified approach to identifying a hypersonic bow shock breaks down when internal shocks or plumes are present in the flow field. However, additional processing of the iso-contour can provide the isolated surface that represents the bow shock. The connectivity of the iso-contour is first analyzed using the *vtkPolyDataConnectivityFilter*. This allows the iso-contour to be broken into separate surfaces, each associated with the various flow phenomena (bow shock, internal shock and plume). Identification of the bow shock is achieved by examining the bounding box for each of the surfaces that make up the iso-contour. For hypersonic conditions, the bow shock will encompass all of the other flow features, so the bounds of the bow shock will match the bounds of the iso-contour as a whole. This approach is particularly efficient since the bounding box information is automatically computed when the VTK data structures are created.

Figure 4 is an example of a complex case with internal flow features. This is the Orion Exploration Flight Test (EFT)-1 launch configuration. The Launch Abort Vehicle (LAV) that surrounds the Orion capsule during ascent has large cavities and protrusions that create numerous internal shocks. This case was chosen because it's currently not possible to perform shock adaption on this geometry using either DPLR or LAURA. Figure 4a shows the center-line Mach iso-contour results for the initial solution. In figure 4b, the initial step of the shock detection process can be seen, as well as the internal flow features. The final results of the shock detection process where the internal flow features have been removed using the outlined process are apparent in figure 4c.



(a) Center-line Mach iso-contour.

(b) Center-line Mach iso-contour with a Mach iso-surface set to 95% of the freestream Mach number and internal features present.

(c) Center-line Mach iso-contour with a Mach iso-surface set to 95% of the freesteam Mach number and internal features removed.

Figure 4. Iso-contour results for a complex case.

III.C. Shock Smoothing

The overall goal of the shock-fitting process is to create a grid with element faces that are aligned to the shock, decreasing the numerical noise. The raw shock surface identified in the prior step can be quite noisy, particularly when the shock lies within a region of tetrahedrons, as seen in figure 3. This noise is eliminated by smoothing the raw shock surface so that it is aligned to the physical shock.

The smoothing algorithms used for structured shock-fitting depend on the structured nature of the grids and aren't directly applicable to unstructured surfaces. Therefore, different algorithms had to be identified for the present work. There are numerous algorithms[16–18] available for smoothing unstructured grids which were studied.

III.C.1. Algorithms

Two of the algorithms studied are based on a diffusion process based on Eq. (1), where X represents the polyhedral surface mesh, λ is a scale factor and $L(X)$ represents the Laplacian.

$$\frac{\partial X}{\partial t} = \lambda L(X) \tag{1}$$

Gaussian smoothing was one of the algorithms studied that is based on this diffusion process. The Laplacian in Eq. (1) is approximated by Eq. (2), where m is the number of neighbors, $N_1(i)$ are the first ring neighbors and x represents the vertex positions. While this algorithm reduces the noise present in the mesh, the side effect is that the mesh will shrink with each iteration.

$$L(X) = \frac{1}{m} \sum_{j \in N_1(i)} x_j - x_i \tag{2}$$

The second diffusion-based algorithm studied was Taubin[16] smoothing. This algorithm consists of two successive Gaussian iterations, the first with a positive factor λ and the second with a negative scale factor μ, greater in magnitude than the first scale factor ($0 < \lambda < -\mu$). Together, the two passes act as a low pass filter, eliminating large changes in curvature while preserving the smaller changes in curvature that define the general mesh shape, minimizing the mesh shrinkage. The downside to this algorithm is that numerous iterations are often required and the effects tend to plateau quickly.

Mean curvature flow[17] was another algorithm studied. As part of the curvature flow family of algorithms, mean curvature flow works by moving each vertex along the point normals by a speed equal to the surface curvature κ, as shown in Eq. (3). This family of algorithms perform better at eliminating low frequency noise than the Gaussian and Taubin algorithms. The downside to using curvature flow is that small time steps must be chosen to prevent the algorithm from becoming unstable.

$$\frac{\partial x_i}{\partial t} = -\kappa_i \mathbf{n_i} \tag{3}$$

There are many definitions of curvature, but for the present work, the mean curvature $\bar{\kappa}$, defined as the average of the principal curvatures, κ_1 and κ_2 (see Eq. (4)), was used exclusively. This was due to mean curvature being easily computed in the VTK framework by applying the *vtkCurvatures* filter.

$$\bar{\kappa} = \frac{1}{2}(\kappa_1 + \kappa_2) \tag{4}$$

The last algorithm studied was the Two-Step smoothing algorithm.[18] The first step averages the face normals, while the second step adjusts the vertex positions to fit the averaged normals. This process is iterated, typically 30-50 times.

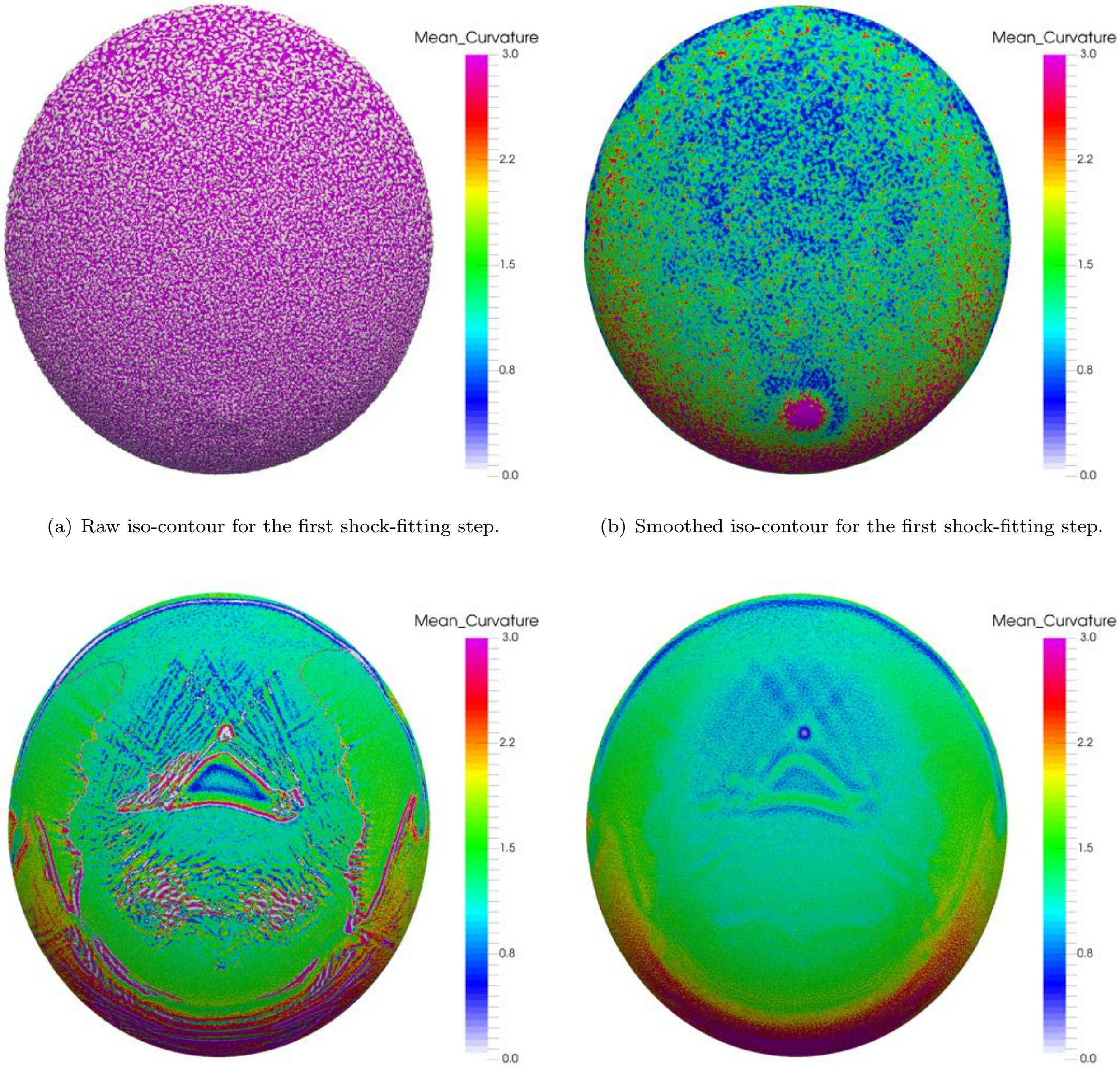
III.C.2. Implementation

The Gaussian, Taubin and mean curvature flow algorithms were all implemented inside of the *mesh_tools* suite, in an explicit manner. Due to its complexity, the Two-Step smoothing algorithm was not implemented internally. Instead, since the algorithm has already been implemented as part of Meshlab,[19] the *mesh_tools* suite calls Meshlab externally when the algorithm is required.

III.C.3. Testing

All of the above algorithms were tested to identify a robust method of smoothing the shock surface. During each test, the mean curvature of the shock surface was computed to evaluate the effectiveness of the smoothing process. The surface curvature is an excellent criteria for evaluating the shock smoothing effectiveness because the enthalpy change across the shock is related to the shock angle. If there is noise in the surface curvature, then the enthalpy downstream of the shock exhibits a proportional amount of noise, leading to erroneous heat flux predictions at the walls.

Testing on a variety of cases found that a combination of algorithms applied in passes was the most robust approach. No single algorithm was able to provide a sufficiently smoothed surface. Gaussian smoothing had the unwanted side effect of shrinking the mesh, but was good at removing particularly noisy regions. Taubin smoothing was perhaps the most conservative algorithm, as it did not shrink the surface, but its effects tended to plateau as successive iterations were applied. The mean curvature flow algorithm provided the smoothest surface with the fewest number of iterations and without any shrinkage, but became unstable easily. The Two-Step smoothing algorithm was the superior method for cleaning particularly noisy surfaces, but had the tendency to introduce noise in smoother surfaces.



(a) Raw iso-contour for the first shock-fitting step.

(b) Smoothed iso-contour for the first shock-fitting step.

(c) Raw iso-contour for the final shock-fitting step.

(d) Smoothed iso-contour for the final shock-fitting step.

Figure 5. Mean curvature results for the axisymmetric heatshield forebody.

III.C.4. Final Smoothing Process

After the shock detection process, the filter *vtkCurvatures* is applied to compute the mean curvature. Based on the amount of noise present, one of two possible routines will be used. For noisy surfaces, such as when the shock lies in a field of tetrahedrons, the best combination was found to be a series of four passes, with each pass consisting of 20 initial Taubin iterations, followed by three iterations of the Two-Step smoothing algorithm and five Gaussian smoothing iterations. For smoother surfaces, the best combination was found to be successive passes with 25 Taubin iterations followed by 80 mean curvature smoothing iterations and five iterations of Gaussian smoothing. Both the Taubin and Gaussian algorithms stabilize the mean curvature iterations.

After each pass, the mean curvature is re-computed and the scripts evaluate whether the most recent pass sufficiently improved the result. If no significant improvements in the surface quality are identified, the resulting surface is passed onto the next step in the process. If the quality is still improving significantly, the process is continued up to a maximum of 30 passes.

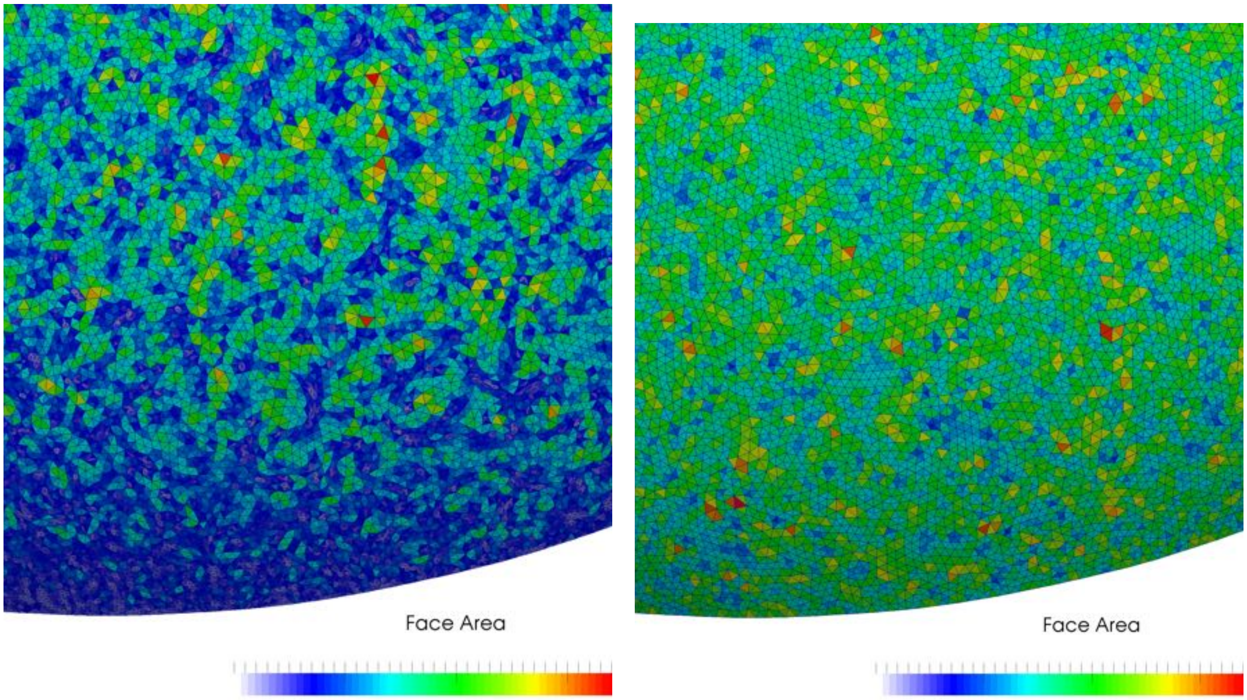
Results of the final smoothing process for the axisymmetric heatshield forebody can be seen in figure 5. For the initial shock-fitting step where the shock lies in the tetrahedral region, the surface has significant noise, apparent by large maximum and minimum values of mean curvature in figure 5a. The smoothing process is then applied and the results in figure 5b show that the noise in the mean curvature is significantly reduced. Also evident in the smoothed surface is a carbuncle in the stagnation region. This indicates that the smoothing process preserves lower frequency features while eliminating the higher frequency noise.

In figure 5c, the raw iso-contour results are shown for the final shock-fitting step. There are still some regions of high and low mean curvature values, where the shock steps between the prismatic layers, but overall, the shock surface is much improved from the initial solution. Additionally, it can be seen that the large carbuncle has been removed during the intermediate shock-fitting steps. The final result of the shock fitting process in figure 5d is now significantly smoother, which will lead to better predictions of the surface heat fluxes. While some noise is still present in the surface, the ratio of the noise to the underlying surface curvature is much less.

III.D. Re-Meshing the Shock Surface

The smoothing process creates an unstructured surface that has minimal noise in the mean curvature but doesn't take into account the element sizing distribution. The smoothing process tends to create face sizes that lead to poor volume element quality. To get the best results from the unstructured shock-fitting process, the smoothed shock surface must be re-meshed to achieve the highest volume element quality possible. To accomplish this, the smoothed shock surface is passed externally to Gmsh,[20] an open source grid generation software. Gmsh then re-meshes the shock surface using a user-defined spacing distribution. Occasionally during the re-meshing process, the surface normals can become flipped. Therefore, after Gmsh has finished, the final shock surface grid is checked to ensure that the surface normals are oriented correctly.

Before and after results of the re-meshing process can be seen in figure 6. Figure 6a is the shock surface after the smoothing process where the faces have various sizes. The results of the re-meshing can be seen in figure 6b, where the final face sizes are distributed much more evenly.



(a) Before re-meshing the surface.

(b) After re-meshing the surface.

Figure 6. Surface mesh face areas before and after re-meshing.

III.E. Extruding a Prismatic Layer around the Shock Surface

Creating elements at the shock surface that have faces aligned with the shock has been demonstrated by Bonfiglioli, et al.[21] to improve the accuracy of the solution. For the present work, it was found that going a step further and creating layers of prismatic elements aligned with the shock provided the best results. The prismatic layer is formed by marching the re-meshed shock surface both upstream and downstream for several layers. The upstream layers are necessary to provide margin for the shock as it settles into the final location throughout successive shock-fitting loops. The number of layers required can depend on the problem, but for the present work 10 to 30 layers were typically used.

The prismatic layers were grown using the Pointwise[22] grid generation package. The *mesh_tools* scripts write out the appropriate Pointwise glyph script and run Pointwise externally. After Pointwise completes the extrusion process, it outputs both the resulting volume grid and the downstream surface of the prismatic shock layer. This downstream surface will be referred to as the shock interface surface for the rest of the paper. A slice though the final volume grid, including the cells around the shock created by the extrusion process, can be seen in figure 7b.

III.F. Rebuilding the Volume Grid

The final step of the unstructured shock-fitting process is to rebuild the volume grid. A new outer boundary is created using the shock interface surface. The shock interface has an open boundary on the downstream side and an exit surface is created to fill the opening and complete the water-tight outer boundary. Depending on the type of problem, the exit surface will either simply fill the hole in the shock interface surface or bridge the space between the viscous walls and the shock interface surface. The completed outer boundary is then merged with the original wall surfaces to complete the boundary surfaces that define the volume. A new volume grid for the region downstream of the shock interface surface is then built in a manner similar to the initial volume grid. Lastly, this volume grid is merged with the prismatic layer around the shock to make the final updated volume grid.

IV. Results

The unstructured shock-fitting methodology outlined was applied to a variety of cases in a production capacity. The *mesh_tools* scripts are not integral to any solver and can be run alongside a solver to complete the process in an automated manner. For the present work however, all of the results presented were performed with the Loci-CHEM solver.

IV.A. Axisymmetric Heatshield Forebody

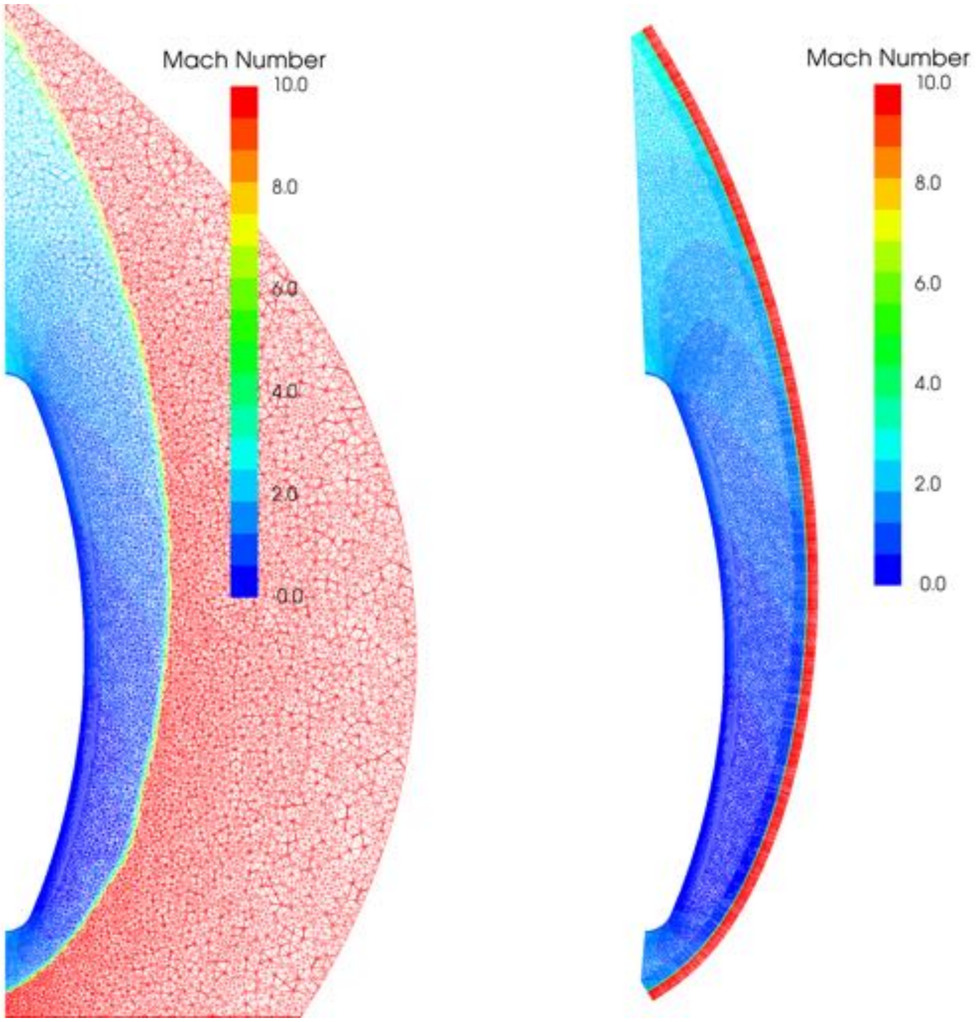
One test case used to validate the unstructured shock-fitting process was the axisymmetric heatshield forebody. This class of problem is solved more efficiently using a structured grid solver, but serves as a basic test of the shock-fitting process.

Validation of the process was performed by making heat flux predictions using the Loci-CHEM solver and comparing the predictions to test data obtained from a series of shock tunnel tests. The data source chosen was the Orion 126CH[23] test run at the Calspan-University at Buffalo Research Center (CUBRC) Lens I shock tunnel. The primary purpose of the test was to define aerothermal environments for the Orion Exploration Mission-1 compression pad design, but there were a number of thin film sensors placed on the vehicle center-line, away from the compression pads, that provide excellent data for the axisymmetric heatshield forebody.

All the 126CH shock tunnel tests were performed with a Mach 10 freestream condition. A number of runs were simulated using the Loci-CHEM solver, but only two representative cases will be shown. Run 9, a low Reynolds number case with purely laminar flow on the forebody and run 8, a high Reynolds number case with turbulent flow over most of the forebody.

IV.A.1. Run 9 - Laminar

The first case was run using laminar conditions, with perfect air and an iso-thermal wall boundary. After obtaining the initial solution, the shock-fitting process was repeated for three loops. Figure 7 shows the center-line Mach iso-contour results for the initial solution and the final shock-fit solution. For the initial solution in figure 7a, the shock lies in a region of tetrahedrons. The noise in the shock surface is apparent in the upper portion of the figure. Additionally, the initial grid is inefficient due to the large number of cells upstream of the shock. The final results in figure 7b show that the shock adaption process has created a prismatic layer of cells, well-aligned with the shock, and that the shock surface is smoother. Another benefit to the shock-fitting process is that only a small portion of the cells lie outside the shock, making the calculations more efficient.



(a) Center-line Mach iso-contour for the initial unstructured grid.

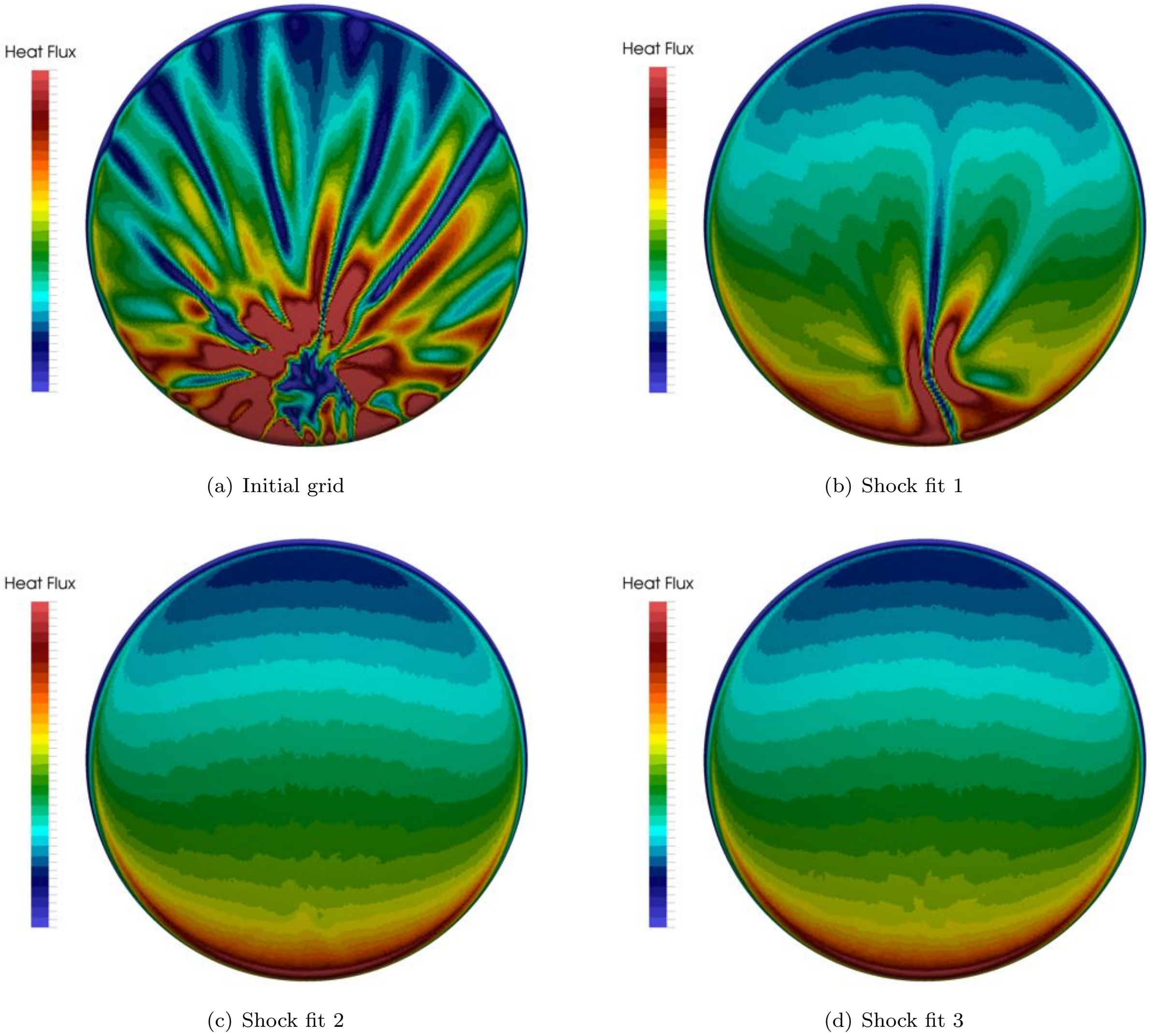
(b) Center-line Mach iso-contour for the final shock-adapted unstructured grid.

Figure 7. Before and after shock-fitting results on an axisymmetric heatshield forebody at Mach 10 using unstructured grids.

The axisymmetric heatshield forebody laminar heat flux results for each shock-fitting iteration can be seen in figure 8. The results for the initial solution in figure 8a show such a high level of noise in the heat flux that the results are completely unusable. A dramatic improvement in the heat flux predictions can be seen in the first shock adaption in figure 8b, due to the addition of the prismatic layer around the shock. Subsequent shock-fitting steps in figures 8c-d show further improvement in the heat flux predictions.

It can also be seen in figure 8 that the shock-fitting process outlined in the present work has some room for improvement. The predicted heat flux isn't completely free of noise, particularly around the stagnation point. Closer inspection of the results suggests this is due to a residual amount of noise present in the shock surface, which can be seen in figure 5d. Further improvements in the shock smoothing process should improve the heat flux predictions.

The predicted center-line heat flux using the shock-fitting process is compared to the run 9 test data and the center-line heat flux predicted by DPLR in figure 9. As expected, the DPLR predictions match



(a) Initial grid

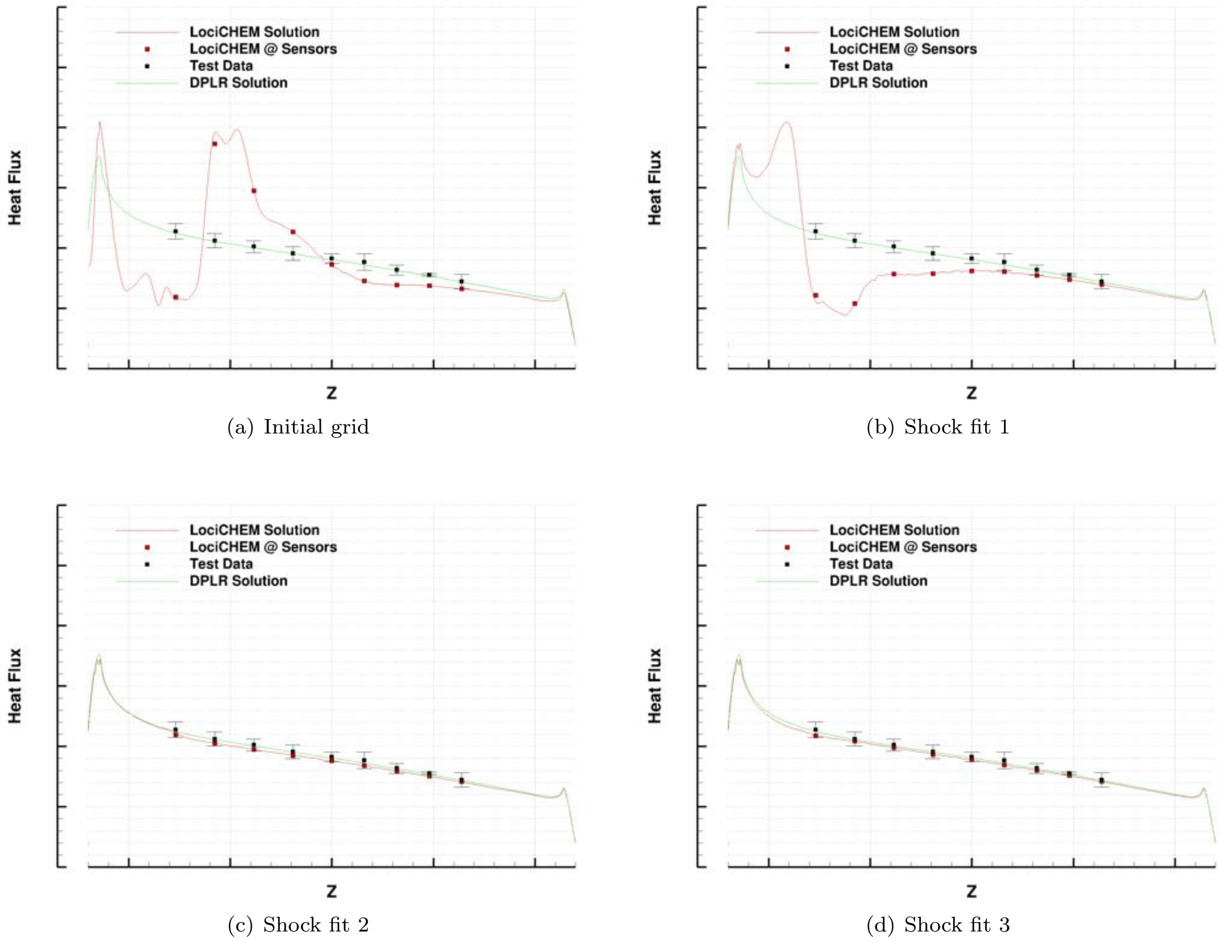
(b) Shock fit 1

(c) Shock fit 2

(d) Shock fit 3

Figure 8. Surface heat flux results for the axisymmetric heatshield forebody at Mach 10 using unstructured grids.

the test data well within the error bars. The Loci-CHEM results using the shock-fitting process follow the trends seen in figure 8, with the initial solution and first shock-fit displaying large variations in heat flux compared to the test data and the DPLR solution. For the second shock-fitting step, shown in figure 9c, the Loci-CHEM predictions correlate with the test data, where all but one of the predictions fall within the error bars. The third and final shock-fitting in figure 9d reveals little change.



(a) Initial grid

(b) Shock fit 1

(c) Shock fit 2

(d) Shock fit 3

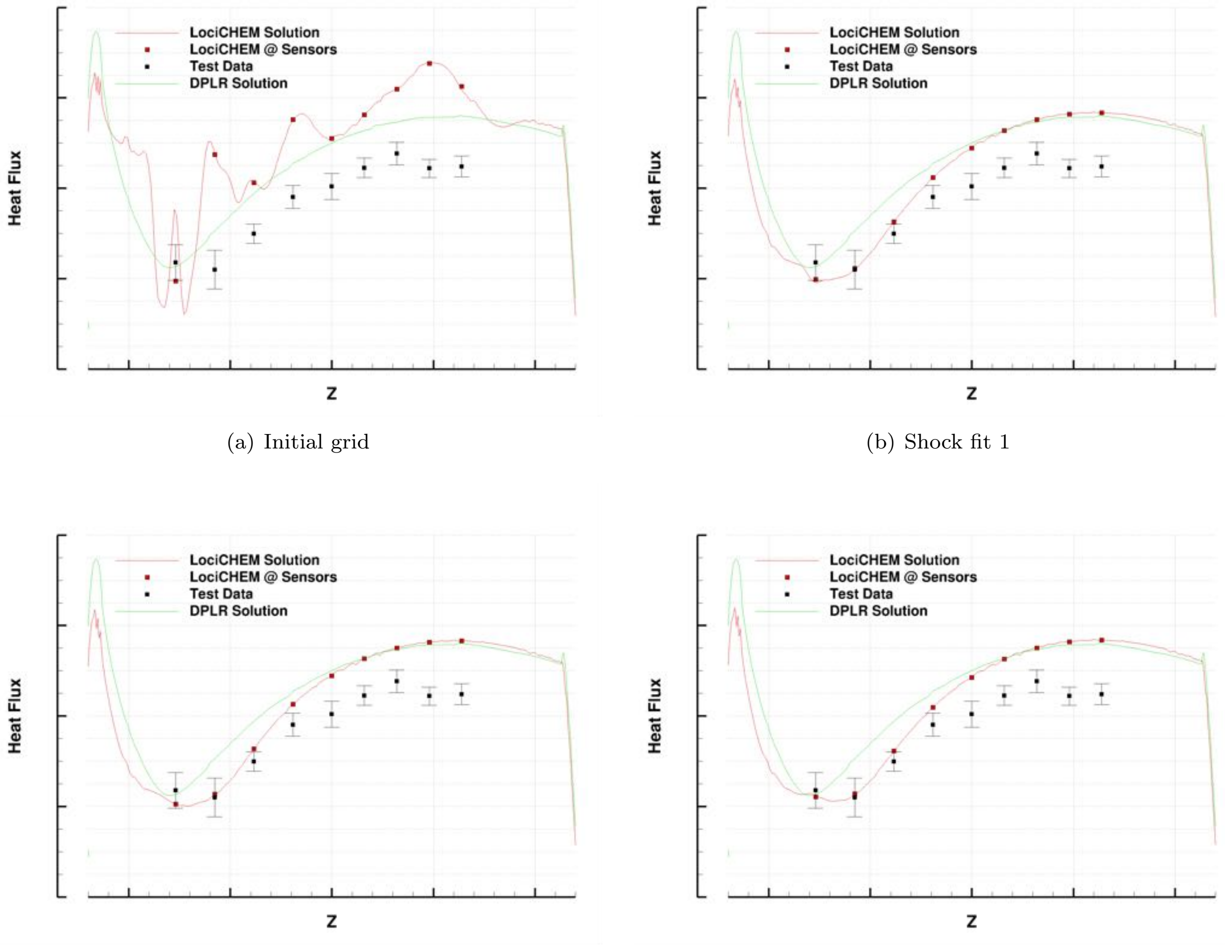
Figure 9. Laminar center-line heat flux results for the axisymmetric heatshield forebody using unstructured grids.

IV.A.2. Run 8 - Turbulent

The second case was run using similar settings as the laminar case but with the Spalart-Allmaras[24] turbulence model enabled. Three shock-fitting iterations were performed for this case.

Figure 10 shows the center-line heat flux results for each of the iterations, along with the test data and DPLR predictions using the Baldwin-Lomax[25] turbulence model. The initial predictions in figure 10a show similar variations, as evidenced in the laminar solutions, but the magnitude of the variations aren't as large, due to the additional diffusion introduced by the turbulence model. This increased diffusion also appears to accelerate the convergence of the shock-fitting process. The Loci-CHEM results for the first shock-fitting step in figure 10b are significantly closer to the DPLR solution than for the first laminar shock-fitting step for run 9. The process has converged for the second and third shock-fitting iterations, with exception to some small variations around the stagnation region. The Loci-CHEM heat flux predictions for the final two iterations match the DPLR solutions for the fully turbulent portion of the heatshield.

It should be noted that while neither turbulence model does a great job of fully matching the test data, the Spalart-Allmaras model does a slightly better job of matching the data where the flow transitions from laminar to turbulent.



(a) Initial grid

(b) Shock fit 1

(c) Shock fit 2

(d) Shock fit 3

Figure 10. Turbulent center-line heat flux results for the axisymmetric heatshield forebody using unstructured grids.

Overall, the laminar and turbulent axisymmetric heatshield forebody cases demonstrate that the shock-fitting process is working as intended, and that the final results correlate with the test data or are consistent with the results obtained from the DPLR solver.

IV.B. Launch Abort Vehicle

The Orion LAV is one of the most complex aerothermal production-level cases due to the irregular geometry and numerous protuberances on the vehicle. Due to its complexity, creating a point-matched structured grid is nearly impossible. Even if possible, neither DPLR or LAURA would be able to successfully shock-fit the grid. The shock-fitting process was run on this geometry, not as a validation case, but as a stressing case, to see if the process could handle it.

For nominal ascent, the Orion LAV rides on top of the launch vehicle to orbit. For EFT-1, the Orion LAV was launched on a Delta IV Heavy. Loci-CHEM was utilized to provide some of the aerothermal environments during nominal ascent. The initial predicted flow field can be seen in figure 4. There are numerous shocks that develop inside of the bow shock as evidenced in figure 4b. Figure 4c shows that the process is able to parse out the internal shocks and isolate the bow shock. The result of one shock-fitting step can be seen in figure 11. The outer boundary now cleanly wraps around the bow shock making the computations more efficient. While the heat flux predictions are not shown, the new topology will lead to improvements in the predicted heat flux.

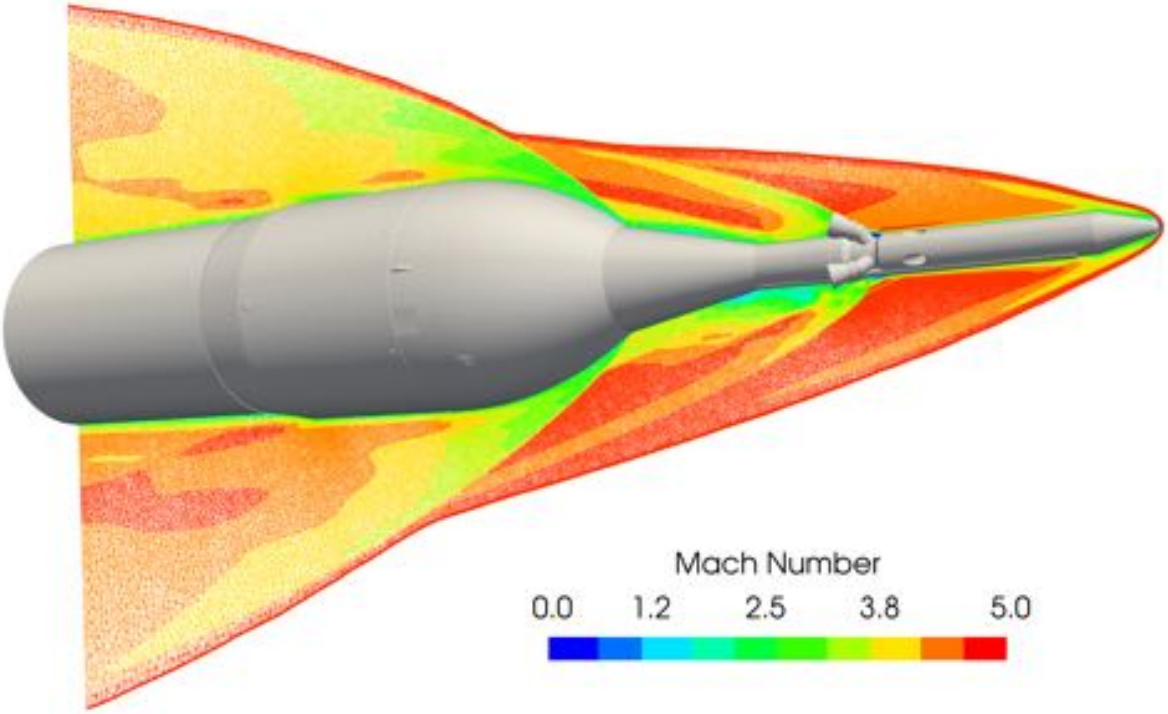


Figure 11. The resulting grid and flow field after performing one shock-fitting iteration for the Orion EFT-1 launch configuration.

V. Conclusion

A process for unstructured shock-fitting to improve unstructured hypersonic CFD predictions has been developed and tested. This process is robust and can handle both internal shocks and plumes that occur with complex geometries. It has been implemented inside of the *mesh_tools* suite and tested alongside the Loci-CHEM CFD solver. The process creates an efficient grid topology with the prismatic layers at the shock and viscous walls, and tetrahedrons in the remaining regions. The results demonstrate a dramatic improvement in the heat flux predictions for unstructured hypersonic problems.

The shock-fitting process does have some aspects that could be improved. Currently, the user must specify the desired Mach number for identifying the shock. An automated method of identifying the correct Mach number would make the script easier to use. The mesh smoothing processes can be improved by implementation of implicit smoothing algorithms to improve their stability.

While re-meshing the volume grid is not computationally expensive, it is a serial process and therefore can take a significant amount of time if the grid is large. The process could be accelerated by moving to a mesh motion process after rebuilding the grid topology during the first iteration.

Acknowledgments

The author would like to thank Ashley Coates for her assistance improving the shock-fitting scripts. Additional thanks go to Daniel G. Schauerhamer for providing valuable feedback on the process.

References

[1] Mystery, A., Sniff, P., Johnston, C., and Kleb, B., *LAURA Users Manual*, Tech. Rep. NASA TM 2010-216836, 2010.

[2] Gnoffo, P., Gupta, R., and Shinn, J., *Conservation Equations and Physical Models for Hypersonic Air Flows in Thermal and Chemical Nonequilibrium*, Tech. Rep. NASA TP-2867, 1989.

[3] Wright, M., *A Family of Data-Parallel Relaxation Methods for the Navier-Stokes Equations*, Ph.D. thesis, University of Minnesota, 1997.

[4] Wright, M., Candler, G., and Bose, D., *Data-Parallel Line Relaxation Method for the Navier-Stokes Equations*, AIAA Journal, Vol. 36, No. 9, 1998, pp. 1603-1609.

[5] Wright, M., White, T., and Mangini, N., *Data-Parallel Line Relaxation (DPLR) Code User Manual Acadia-Version 4.01.1*, NASA TM-2009-215388, NASA Ames Research Center, October 2009.

[6] Boger, D., Noack, R., Amar, A., Kirk, B., Lillard, R., Olsen, M., and Dries, K., *Overset Grid Applications in Hypersonic Flow Using the DPLR Flow Solver*, AIAA-2008-921, 46th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, 2008.

[7] Hyatt, A., Barnhardt, M., Prabhu, D., Mansour, N., and Boger, D., *Verification and Validation of Structured and Overset Grid Capabilities on Hypersonic Multibody Configurations*, AIAA-2010-4454, 40th Fluid Dynamics Conference and Exhibit, Fluid Dynamics and Co-located Conferences, 2010.

[8] Candler, G., Subbareddy, P., and Brock, J., *Advances in Computational Fluid Dynamics Methods for Hypersonic Flows*, Journal of Spacecraft and Rockets, Vol. 52, Special Section on Numerical Simulation of Hypersonic Flows, 2015, pp. 17-28.

[9] Biedron, R., et al., *FUN3D Manual: 12.9*, Tech. Rep. NASA TM 2016-219012, 2016.

[10] Luke, E. and George, T., *Loci: A Rule-Based Framework for Parallel Multidisciplinary Simulation Synthesis*, Journal of Functional Programming, Vol. 15, No. 3, 2005, pp. 477-502.

[11] Luke, E., *On Robust and Accurate Arbitrary Polytope CFD Solvers (Invited)*, 18th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences, AIAA-2007-3956, 2007.

[12] Schroeder, W., Martin, K., and Lorensen, B., *The Visualization Toolkit (4th ed.)*, Kitware, ISBN 978-1-930934-19-1, 2006.

[13] *HyperMesh*, http://www.altairhyperworks.com/product/HyperMesh, Accessed: 2016-05-13.

[14] Marcum, D., *Unstructured Grid Generation Using Automatic Point Insertion and Local Reconnection*, The Handbook of Grid Generation, Edited by J.F. Thompson, B. Soni, and N.P. Weatherill, Chapter 18, CRC Press, 1998, pp. 1-31.

[15] Marcum, D. and Weatherill, N., *Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection*, AIAA Journal, Vol. 33, No. 9, 1995, pp. 1619-1625.

[16] Taubin, G., *A Signal Processing Approach to Fair Surface Design*, Proceedings of ACM SIGGRAPH 95, 1995, pp. 351-358.

[17] Desbrun, M., Meyer, M., Schroder, P., and Barr, A., *Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow*, Proceedings of ACM SIGGRAPH 99, 1999, pp. 317-324.

[18] Belyaev, A. and Ohtake, Y., *A Comparison of Mesh Smoothing Methods*, Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics, Vol. 2, 2003.

[19] Callieri, M., Ranzuglia, G., Dellepiane, M., Cignoni, P., and Scopigno, R., *Meshlab as a Complete Open Tool for the Integration of Photos and Colour with High-Resolution 3D Geometry Data*, CAA 2012 Conference Proceedings, 2013, pp. 406-416.

[20] Geuzaine, C. and Remacle, J., *Gmsh: A Three-Dimensional Finite Element Mesh Generator with Built-In Pre- and Post-Processing Facilities*, International Journal for Numerical Methods in Engineering, Vol. 79, No. 11, 2009, pp. 1309-1331.

[21] Bonfiglioli, A., Grottadaurea, M., Paciorri, R., Sabetta, F., Onofri, M., and Bianchi, D., *Numerical Simulation of Hypersonic Flows Past Three-Dimensional Blunt Bodies Through an Unstructured Shock-Fitting Solver*, 17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, International Space Planes and Hypersonic Systems and Technologies Conferences, 2011.

[22] *Pointwise*, http://www.pointwise.com/pw/, Accessed: 2016-05-13.

[23] *Experimental Studies of a High Fidelity ORION Heat Shield with Compression Pad Geometries: 126CH*, 2015.

[24] Spalart, P.R. and Allmaras, S.R., *A One-Equation Turbulence Model for Aerodynamic Flows*, AIAA-92-0439, 1992.

[25] Baldwin, B. and Lomax, H., *Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows*, Proceedings of the AIAA 16th Aerospace Sciences Meeting, AIAA-78-258, 1978.